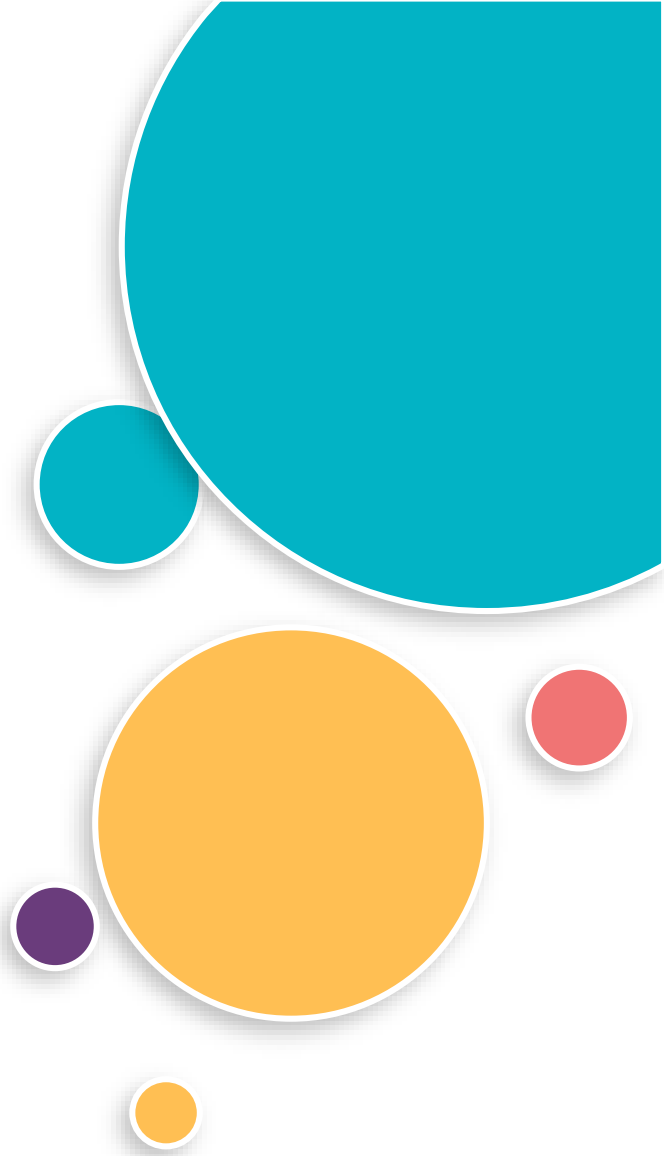
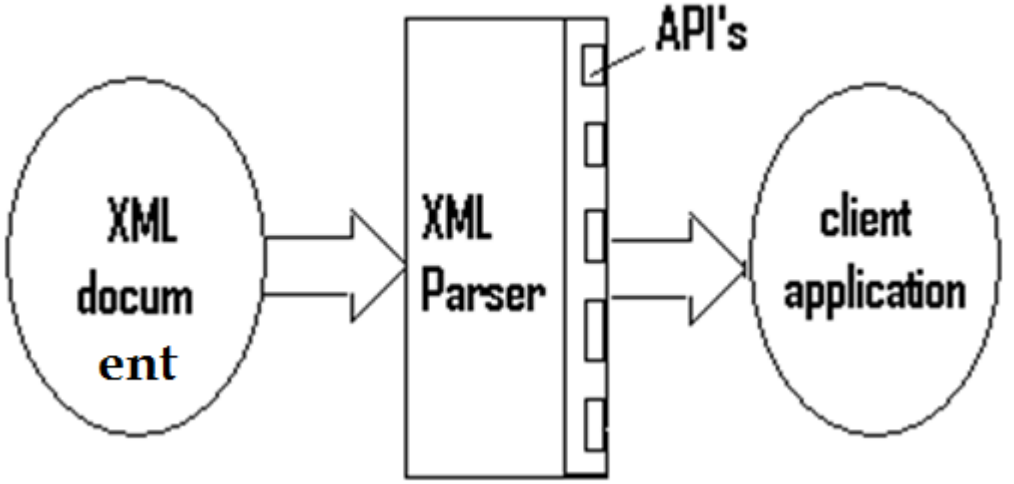
TOC:

- What is a XML Parser
- DOM & SAX parser API
- Xerces-J parsers overview
- Work with XML parsers



What is XML parser?

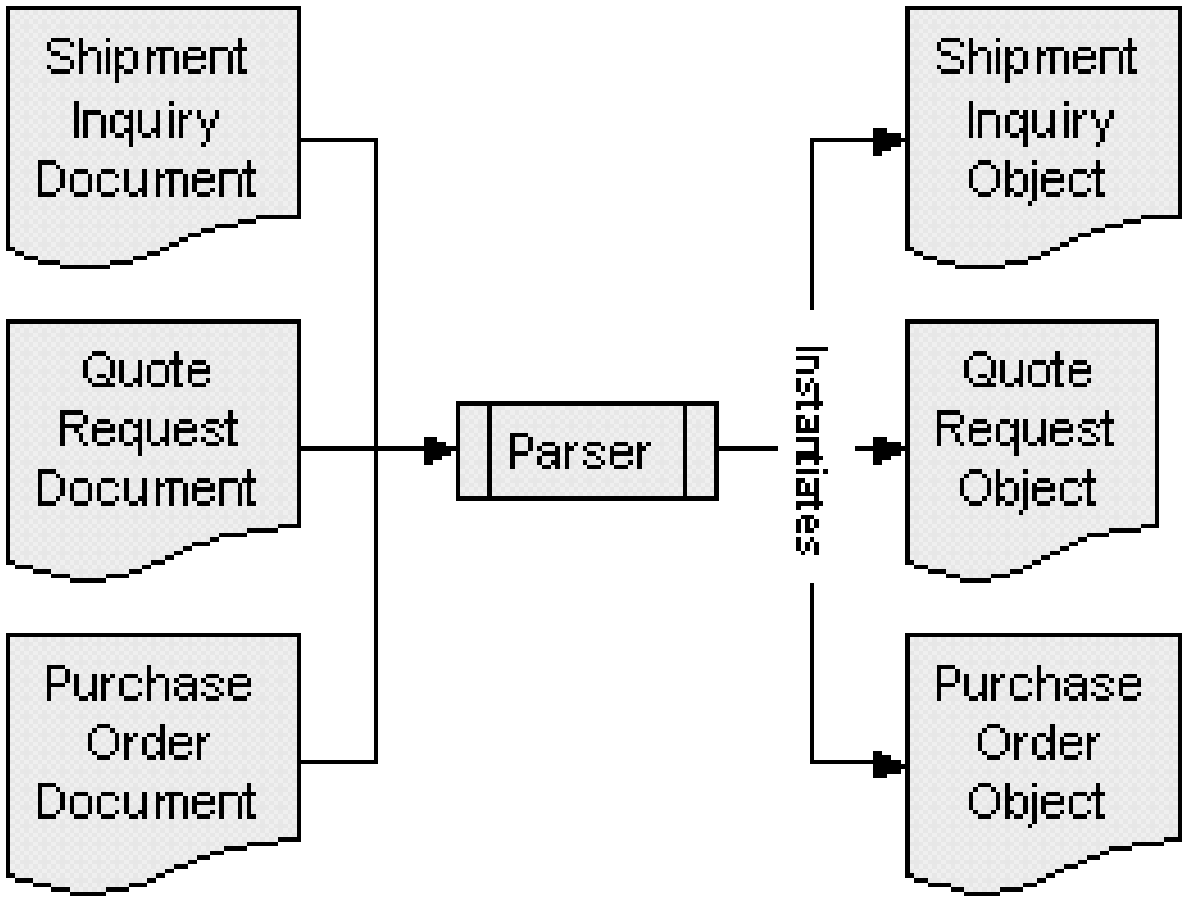
1. An XML parser is a software library (or a package) that provides methods (or interfaces) for client applications to work with an XML document. The XML Parser is designed to read the XML and create a way for programs to use XML.
2. XML Parser validates the document and check that the document is well formatted.
3. It does a lot of other detailed things so that a client is shielded from that complexities.
4. Let’s understand the working of XML parser by the figure:



Types of XML Parsers

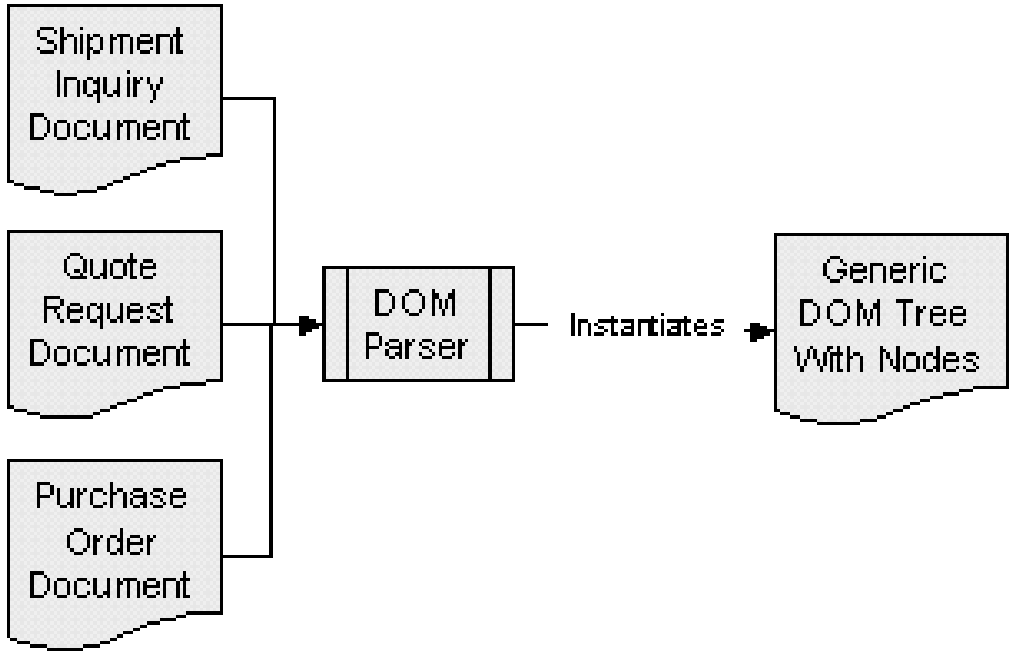
There are two main types of XML parsers:

- DOM (Document Object Model)
- SAX (Simple API for XML)



DOM

A DOM document is an object which contains all the information of an XML document. It is composed like a tree structure. The DOM Parser implements a DOM API defining Objects and properties of all elements and methods (interface) to access them. This API is very simple to use.



Features of DOM parser

- A DOM parser creates an internal structure in memory which is a DOM document object and the client applications gets information of the original XML document by invoking methods on this document object.
- DOM parser has a tree based structure.

DOM

Advantages

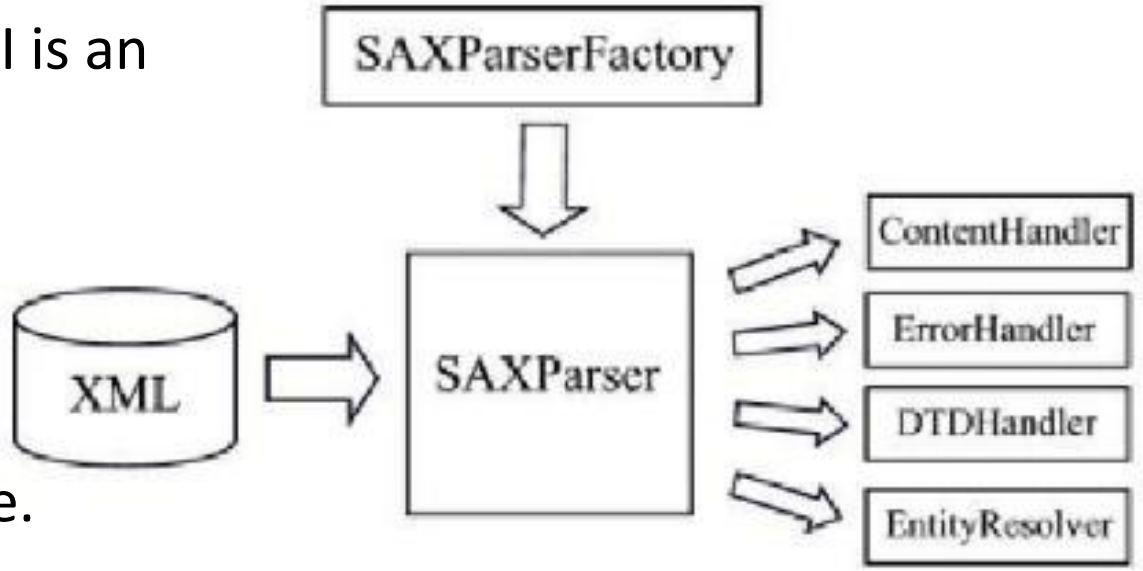
- It supports both read and write operations and the API is very simple to use.
- It is preferred when random access to widely separated parts of a document is required.
- The XML DOM is a standard for **how to get, change, add or delete XML elements.**

Disadvantages

- It is memory inefficient. (Consumes more memory because the whole XML document needs to loaded into memory).
- It is comparatively slower than other parsers.

SAX

A SAX Parser implements SAX API. This API is an event based API and less intuitive.



Features of DOM parser

- It does not create any internal structure.
- Clients does not know what methods to call, they just overrides the methods of the API and place his own code inside method.
- It is an event based parser, it works like an event handler in Java.

SAX

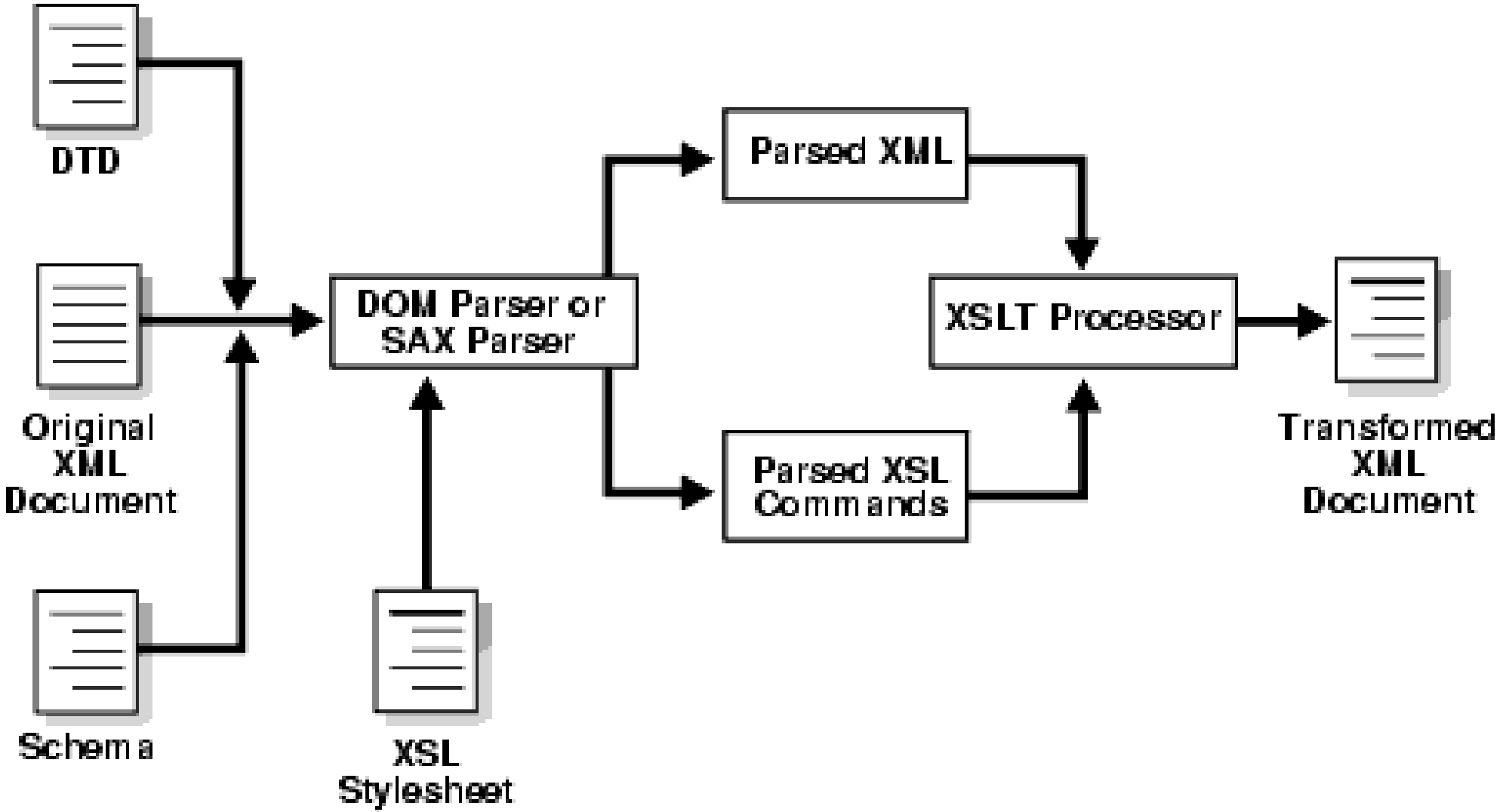
Advantages

- It is simple and memory efficient.
- It is very fast and works for huge documents. Works well in stream application.

Disadvantages

- It is event-based so its API is less intuitive.
- Clients never know the full information because the data is broken into pieces.

Parser API Mechanism



DOM Application

Working with XML Parser

XML-DOM

This example loads a text string into an XML DOM object, and extracts the info from it with JavaScript.

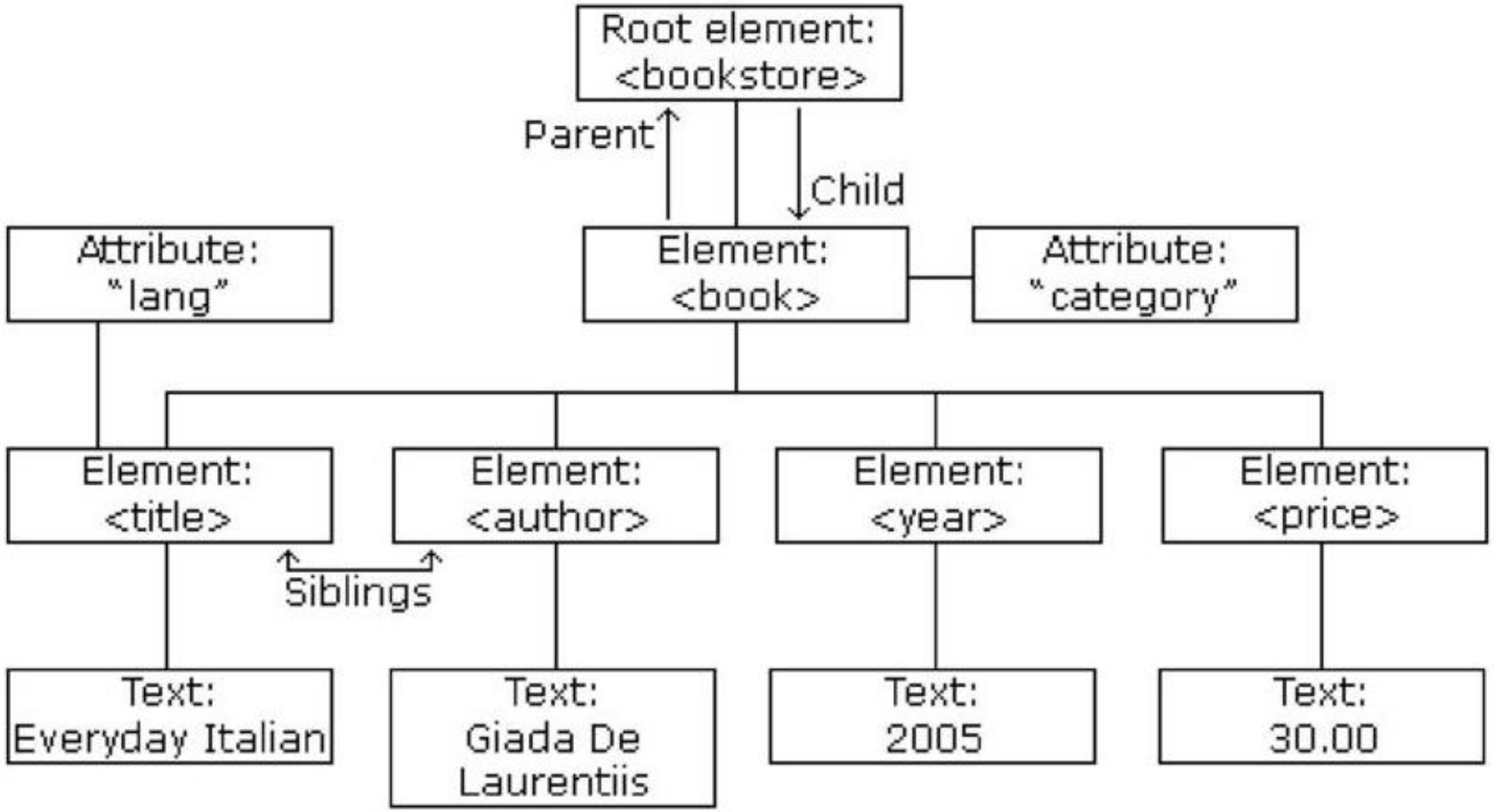
```
<html>
<body>
        <p id="demo"></p>
        <script>
        var text, parser, xmlDoc;
        text = "<bookstore><book>"+
                "<title>Everyday Nepal</title>"+
                "<author>Yubaraj Rai</author>"+
                "<year>2005</year>"+
                "</book></bookstore>";

        parser = new DOMParser();
        xmlDoc = parser.parseFromString(text, "text/xml");
        document.getElementById("demo").innerHTML = xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
        </script>
</body>
</html>
```

Output:

Everyday Nepal

XML DOM
(XML Nodes)



DOM Nodes

- According to the DOM, everything in an XML document is a **node**.
- The entire document is a **document node.**
- Every XML element is an **element node.**
- The text in the XML elements are **text nodes.**
- Every attribute is an **attribute node.**
- Comments are **comment nodes.**

In XML-DOM example:

- The root node <bookstore> holds four <book> nodes.
- Each <book> node contain 4 text node: i.e. <title>, <author>, <year>, <price>

DOM Parsing

- Most browsers have a built-in XML parser to read and manipulate XML.
- The parser reads XML into memory and converts XML into XML DOM object which is accessible from **JavaScript**.
- There are some differences between **Microsoft's XML Parser** and the parsers used in others browsers.
- The Ms. Parser supports **loading of both XML files and XML strings (text).**
- Other browsers use **Separate Parsers**.
- However, all parsers contain functions to **traverse XML tree, access, insert, and delete nodes**.

DOM Properties and Methods

DOM Properties and Methods: (x is node object)

- x.nodeName - the name of x
- x.nodeValue - the value of x
- x.parentNode - the parent node of x
- x.childNodes - the child nodes of x
- x.attributes - the attributes nodes of x

XML DOM Methods:

- x.getElementsByTagName(Name) - get all elements with a specified tagname.
- x.appendChild(Node) - insert a child node to x
- x.removeChild(Node) - remove a child node from x